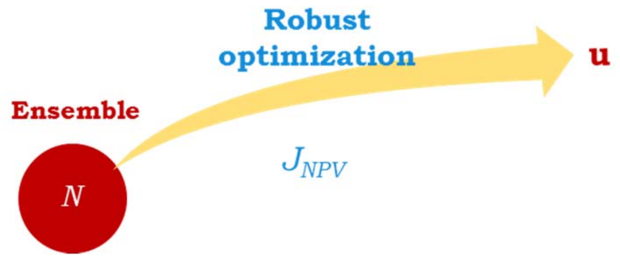
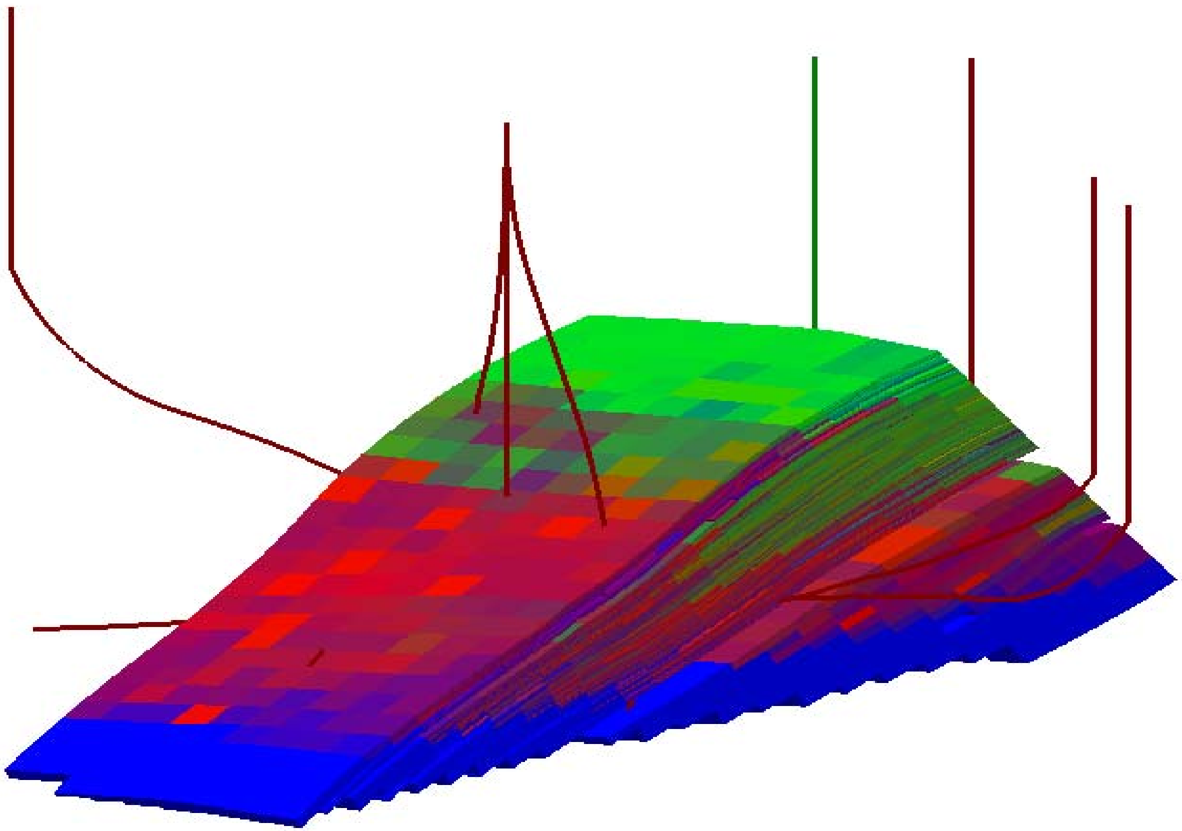
# Assessing the Value of Information from Inverse Modelling for Optimising Long-Term Oil Reservoir Performance

**Eduardo Barros, TU Delft**
**Paul Van den Hof, TU Eindhoven**
**Jan Dirk Jansen, TU Delft**



# Oil & gas reservoirs

fluids trapped in porous rock below impermeable ‘cap rock’



# Notation

System equations: $\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k, \boldsymbol{\theta}\right) = \mathbf{0}\,,\;\; k = 1, 2, \ldots, K$

Output equations: $\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) = \mathbf{0}$

States: $\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \mathbf{s}^T \end{bmatrix}^T$ pressures, saturations

Parameters: $\boldsymbol{\theta} = \begin{bmatrix} \mathbf{k}^T & \boldsymbol{\varphi}^T \end{bmatrix}^T$ perms, porosities, …

Inputs: $\mathbf{u} = \begin{bmatrix} \breve{\mathbf{p}}_{well}^T & \breve{\mathbf{q}}_{well}^T \end{bmatrix}^T$ well pressures, total rates

Outputs: $\mathbf{y} = \begin{bmatrix} \mathbf{p}_{well}^T & \mathbf{q}_{well,o}^T & \mathbf{q}_{well,w}^T \end{bmatrix}^T$ well press., phase rates

# Governing equations – simple example

- Oil and water only, no gravity, no capillary pressures
- Separate equations for $p$ and $S_w$ :

$$-k\lambda_t \nabla^2 p + \phi c_t \frac{\partial p}{\partial t} = q_t$$

diffusion

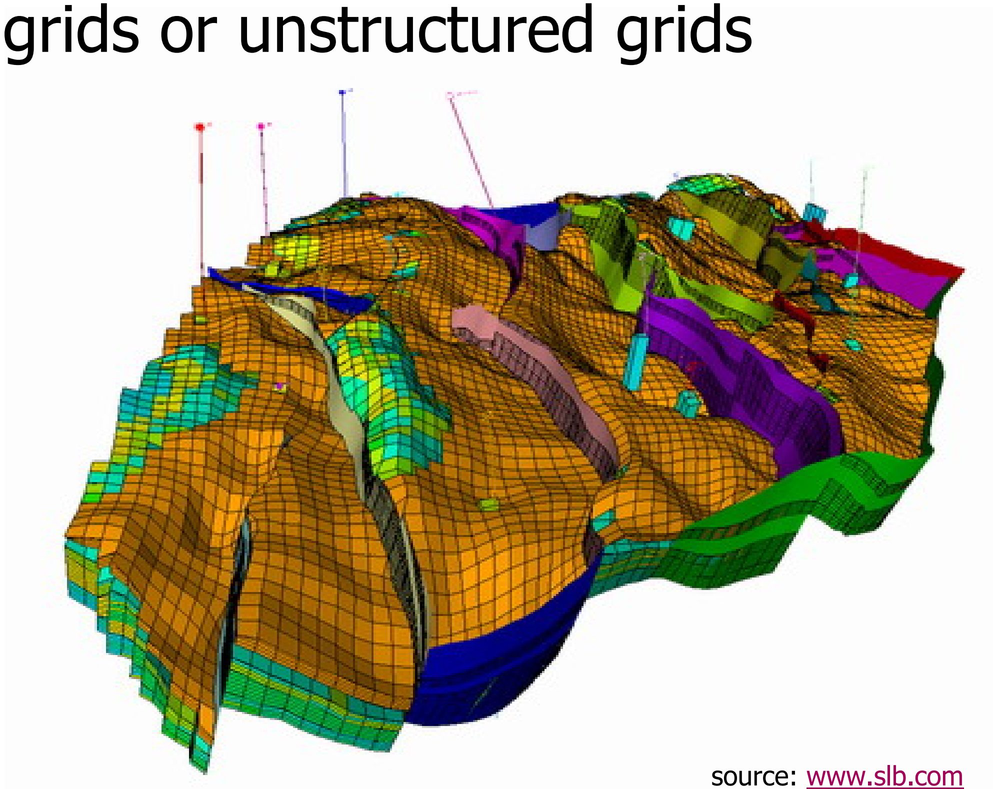
$$v_t \nabla f_w\left(S_w\right) + \phi \frac{\partial S_w}{\partial t} = q_w$$

convection

- $\lambda_t$, $c_t$ and $f_w$ are functions of $S_w$; $v_t$ is a function of $p$
- Coupled and nonlinear, (near-)elliptic, (near-)hyperbolic

# Reservoir simulation

- 3-phases (gas, oil, water) or multiple components
  + thermal effects + chemical effects + geo-mechanics + …
- Nonlinear PDEs discretized in time and space – FD/FV
- Cornerpoint grids or unstructured grids



source: www.slb.com

# Reservoir simulation

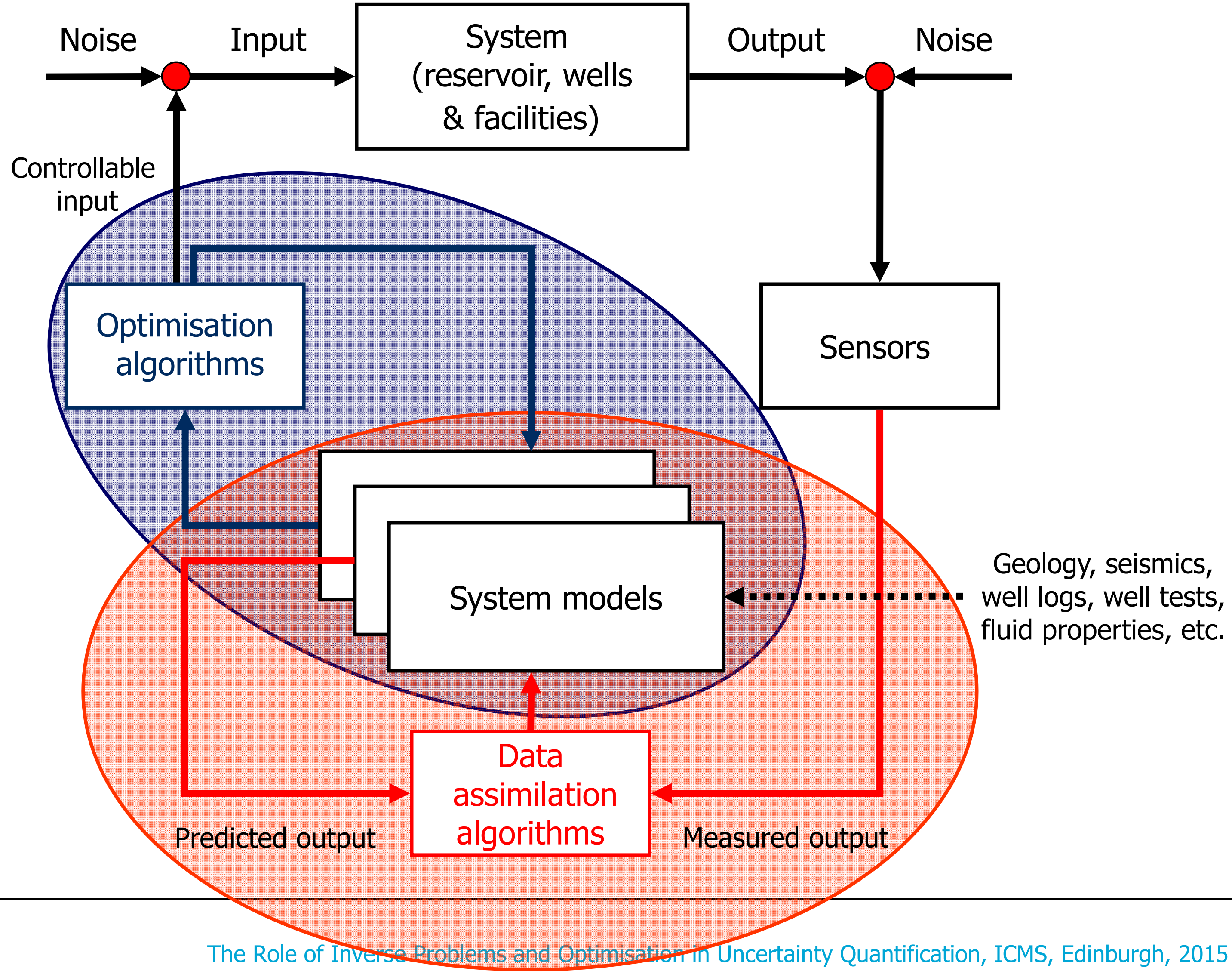
- 3-phases (gas, oil, water) or multiple components
  + thermal effects + chemical effects + geomechanics + …
- Nonlinear PDEs discretized in time and space – FD/FV
- Cornerpoint grids or unstructured grids
- Large variation in parameter values: $10^{-15} < k < 10^{-11}$ m$^2$
- Typical model size: $10^4$–$10^6$ cells, 50–500 time steps
- Fully implicit (Newton iterations) – clock times: hours-days
- Typical code size: $10^6$-$10^7$ lines (well models, PVT analysis)
- Research focused on upscaling, gridding, 'history matching' (inverse modeling), new physics, solvers, parallelization
- Primarily used in design phase: field (re-)development

Reservoir simulation models are used in 'batch mode'

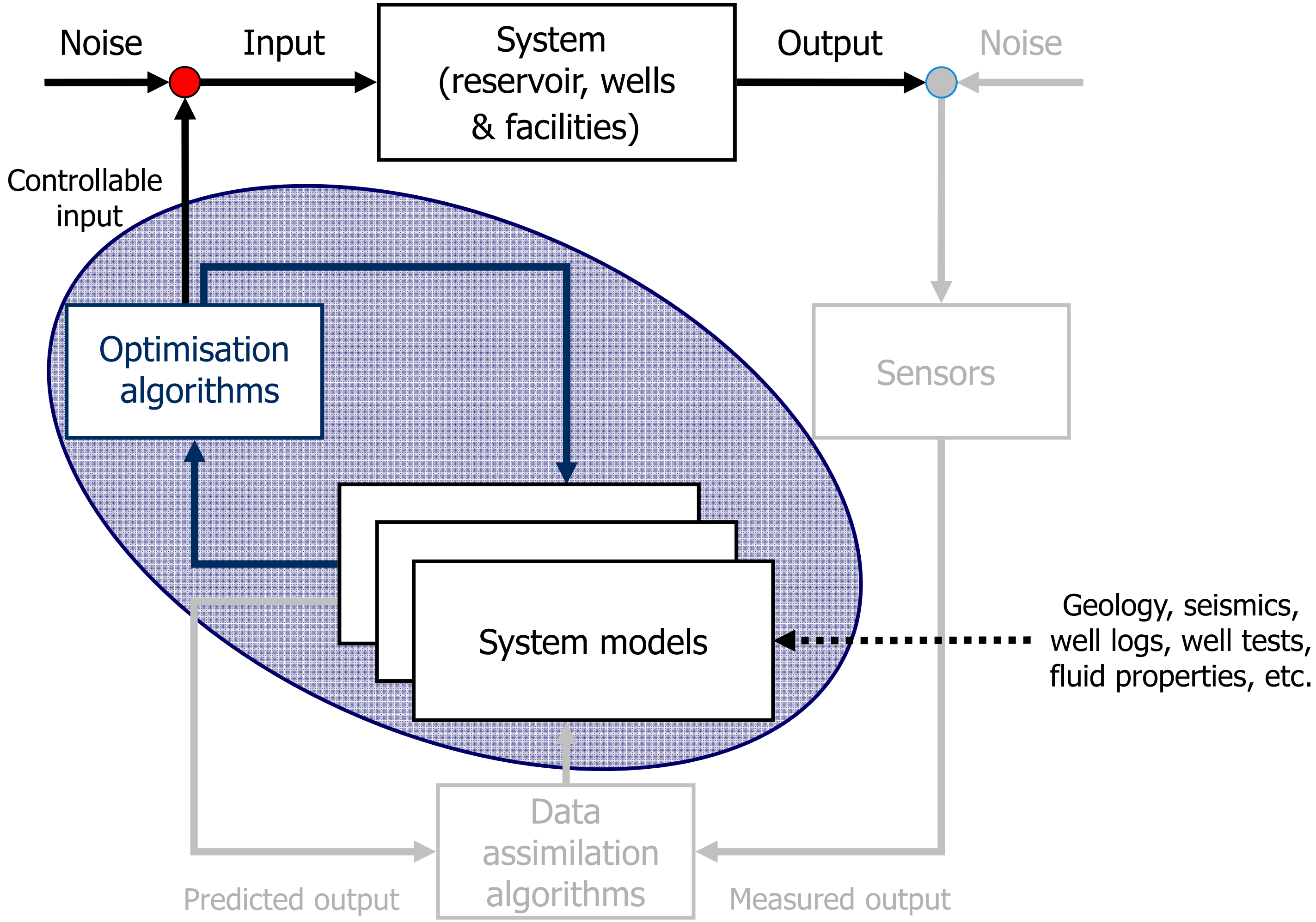
# Closed-loop reservoir management

- Hypothesis: recovery can be significantly increased by changing reservoir management from a ‘batch-type’ to a near-continuous model-based controlled activity
- Key elements:
  - Optimization under geological uncertainties
  - Data assimilation for frequent updating of system models
- Inspiration:
  - Systems and control theory
  - Meteorology and oceanography
- A.k.a. real-time reservoir management, smart fields, intelligent fields, integrated operations, …

# Closed-loop reservoir management



# Robust flooding optimisation



# Robust flooding optimisation

- problem statement: $\max_{\mathbf{u}_{1:K}} \frac{1}{N} \sum_{i=1}^{N} J_i\left(\mathbf{u}_{1:K}, \mathbf{y}_{1:K}, \boldsymbol{\theta}_i\right)$ subject to
- system equations: $\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0}$
- initial conditions: $\mathbf{x}_0 = \breve{\mathbf{x}}_0$
- output equation: $\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) = \mathbf{0}$
- equality constraints: $\mathbf{c}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) = \mathbf{0}$
- inequality constraints: $\mathbf{d}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) < \mathbf{0}$, $\quad k = 1, 2, \ldots, K$

100s of geological realizations

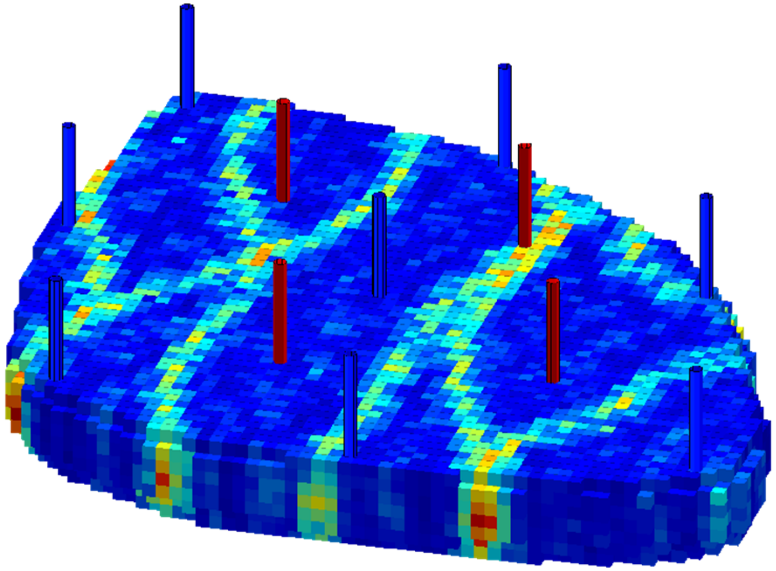
10 wells, 100 time steps => 10000 optimization parameters

# Optimisation techniques

- Global versus local
- Gradient-based versus gradient-free
- Constrained versus non-constrained
- ‘Classical’ versus ‘non-classical’ (genetic algorithms, simulated annealing, particle swarms, etc.)
- We use ‘adjoint-based optimal control theory’
  - Gradient-based – local optimum
  - Computational effort independent of number of controls
  - Objective function: ultimate recovery or monetary value
  - Controls: injection/production rates, pressures or valve openings
  - Beautiful, but code-intrusive and requires lots of programming

Anyway, the magic isn’t in the method

# 12-well example



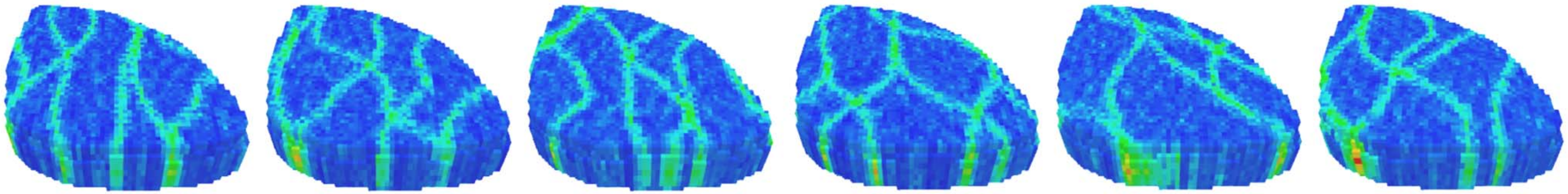
Van Essen et al., 2009

- 3D reservoir
- High-permeability channels
- 8 injectors, rate-controlled
- 4 producers, pressure-controlled
- Production period of 10 years
- 12 wells x 10 x 12 time steps
  gives 1440 optimization parameters
- Optimisation of Net Present Value (NPV) $J$

$J$ = (value of oil – costs of water produced/injected)

# Robust optimisation

- Use ensemble of geological realisations (typically 100)

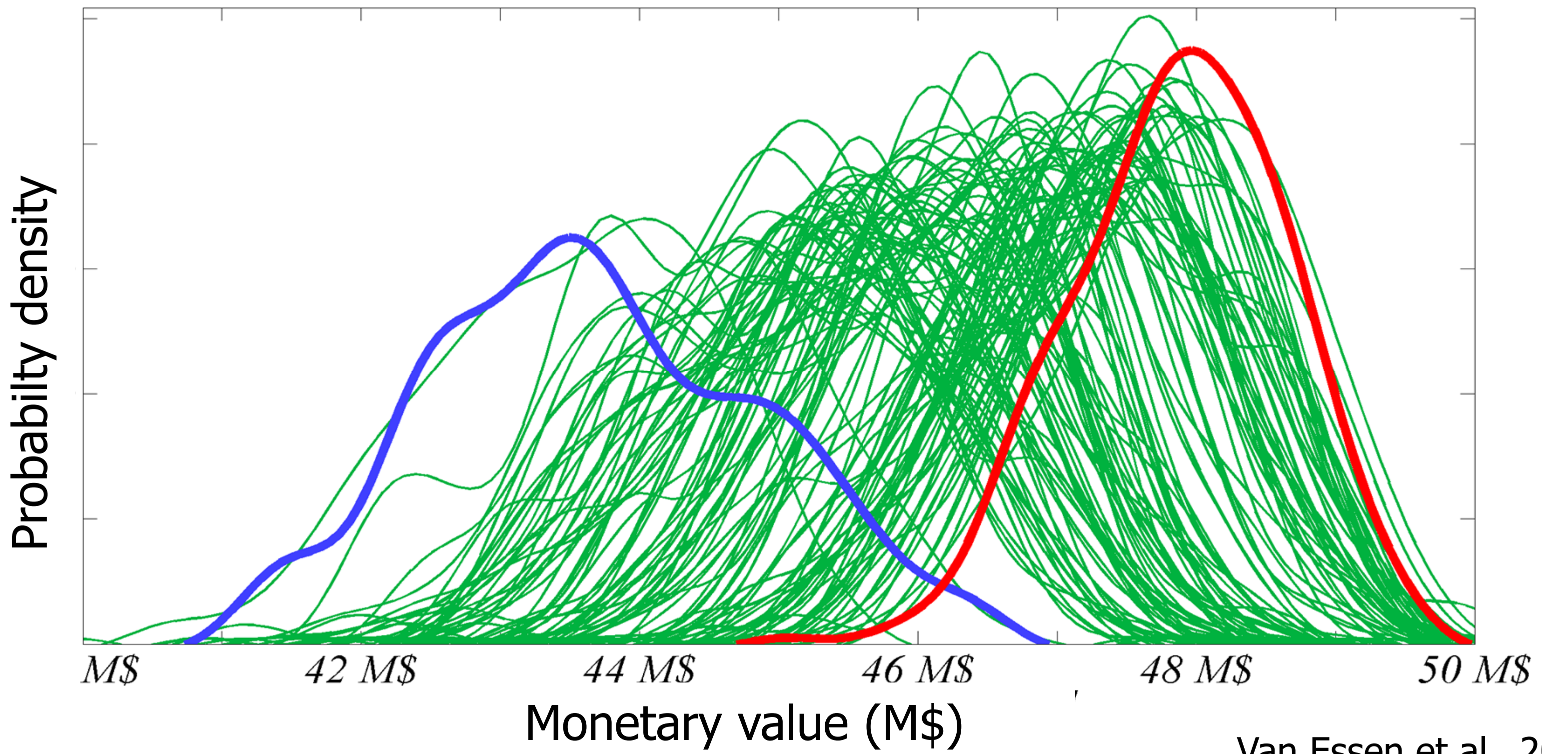


Van Essen et al., 2009

- Optimise expected value over ensemble
- Single strategy, not 100!
- If necessary include risk aversion (utility function)
- Computationally intensive

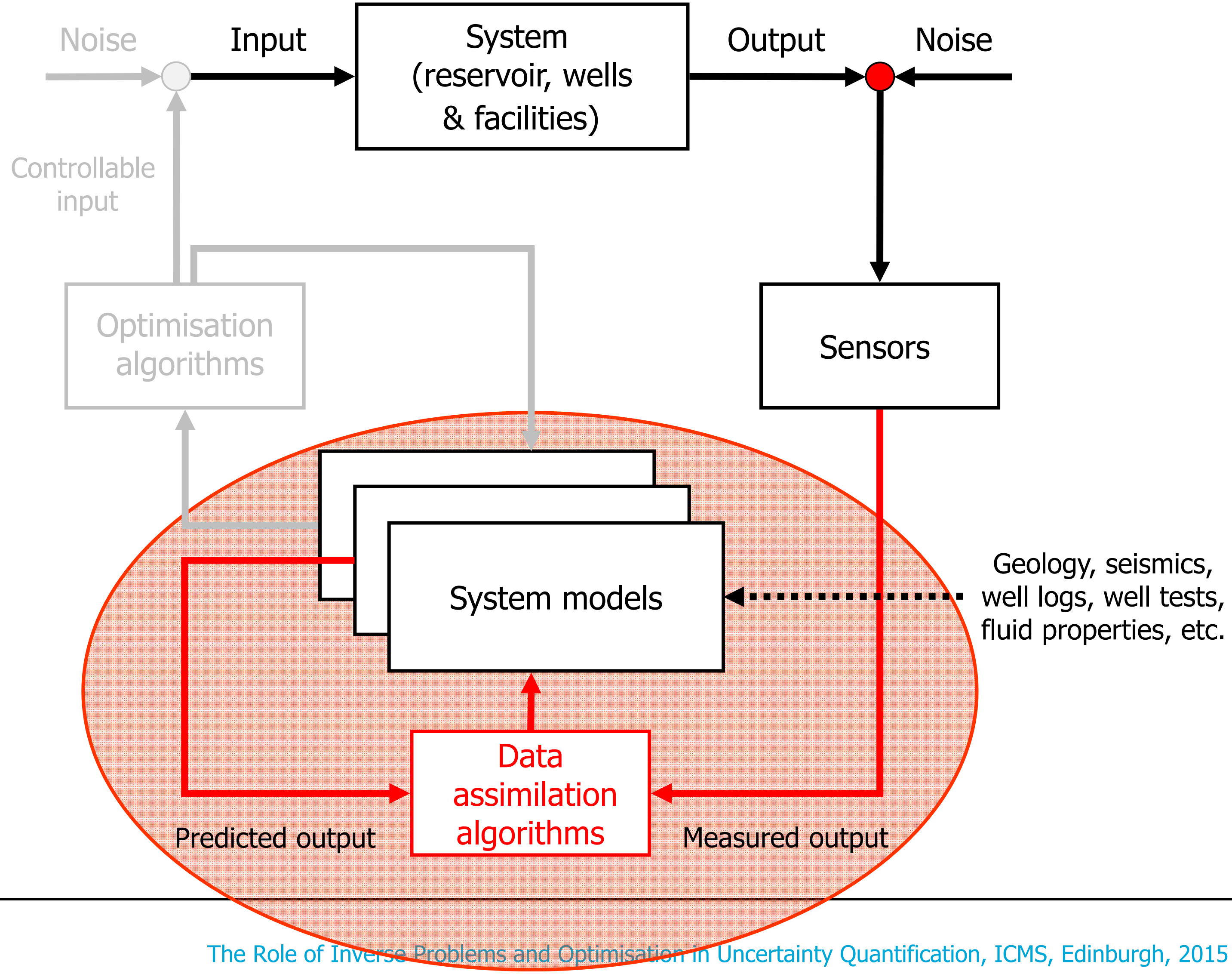
# Robust optimisation results

3 control strategies applied to set of 100 realisations:
reactive control, nominal optimisation, robust optimisation




Van Essen et al., 2009

# ‘Computer-assisted history matching’



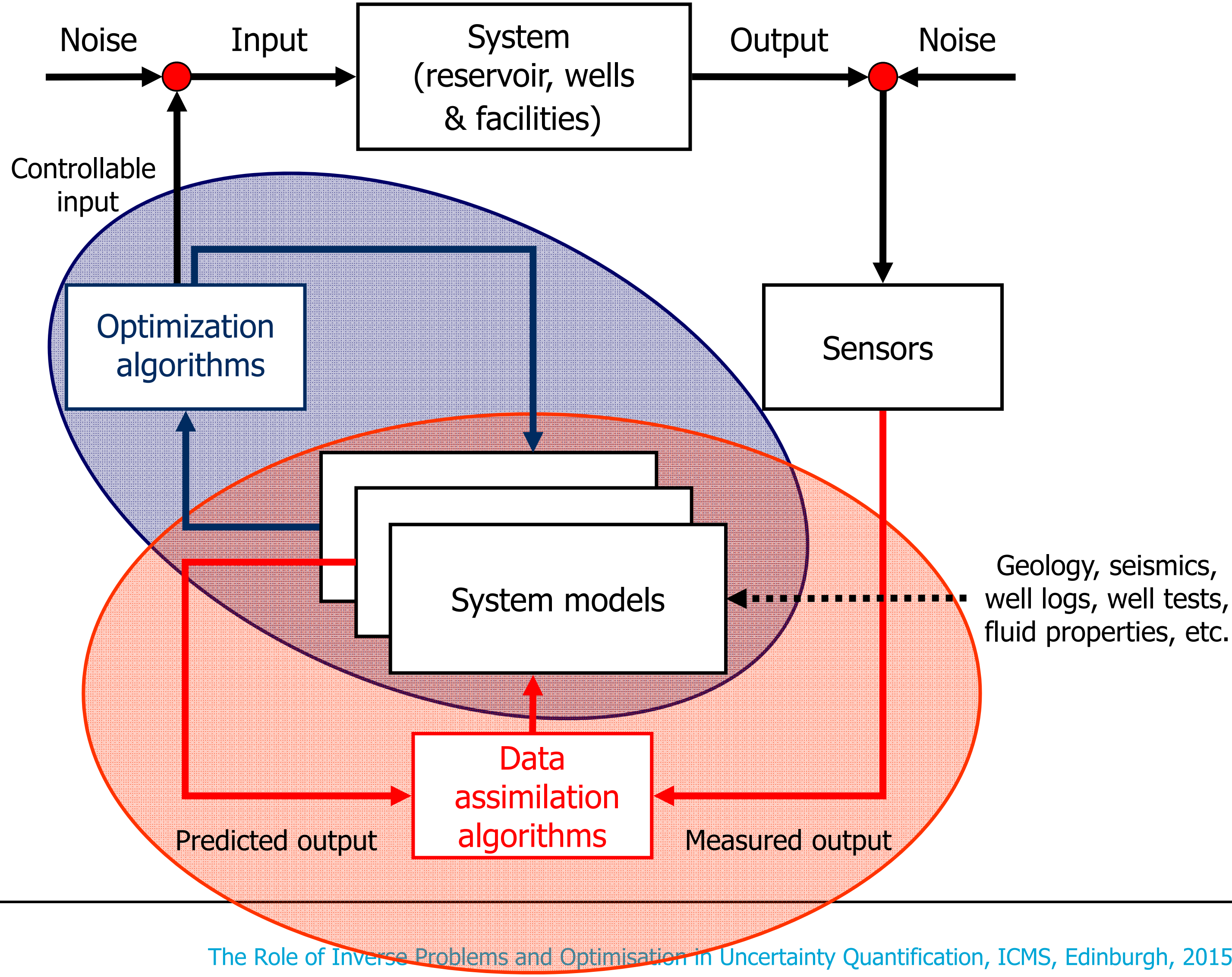
# ‘Computer-assisted history matching’ (Data assimilation/inverse modelling)

- Uncertain parameters, not initial conditions or states
- Parameters: permeabilities, porosities, fault multipliers, …
- Data: production (oil, water, pressure), 4D seismics, …
- Very ill-posed problem: many parameters, little info
- Variational methods – Bayesian framework:

$$J = \left(\mathbf{d}-\mathbf{y}\right)^T \mathbf{P}_d^{-1}\left(\mathbf{d}-\mathbf{y}\right) + \left(\mathbf{m}-\mathbf{m}_0\right)^T \mathbf{P}_m^{-1}\left(\mathbf{m}-\mathbf{m}_0\right)$$

- Ensemble Kalman filtering – sequential methods
- Reservoir-specific methods (e.g. streamlines)
- ‘Non-classical’ methods – simulated annealing, GAs, …
- Monte Carlo methods – MCMC with proxies

# How to assess VOI in CLRM?

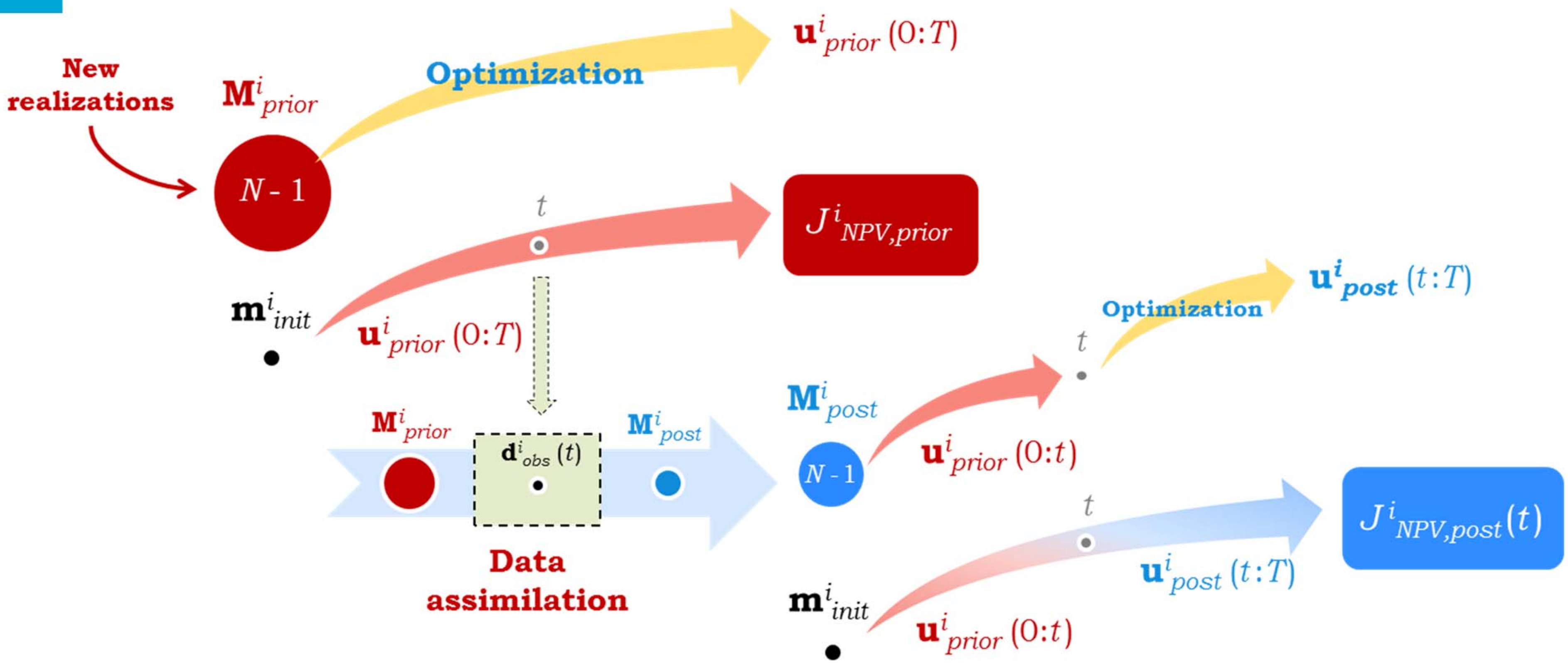


# Workflow to assess VOI in advance (1)

- Generate initial ensemble of geological realizations
- Select one member as 'truth' and create synthetic data
- Generate new ensemble and perform robust optimization
- Rerun resulting strategy on 'truth' and compute NPV
- Perform CLRM using synthetic data
- Rerun resulting strategy on 'truth' and compute $\Delta$ NPV
- Repeat for different synthetic 'truth's (loop over ensemble)
- Compute VOI as average of $\Delta$ NPVs

# Workflow to assess VOI in advance (2)

- For ensemble member $i$ :



# Other measures and notes

- Value Of Clairvoyance (VOC): VOI under sudden and complete revelation of the truth (VOI ≤ VOC) (VOC is upper bound to VOI; 'technical limit')

- Chance Of Knowing: COK = VOI/VOC  (0 ≤ COK ≤ 1)

- Note 1: VOI has only meaning in a decision context; here the decision is the life-cycle production optimisation

- Note 2: Measurements costs are not taken into account. Data from an individual realisation may result in reduction in NPV. On average, data should always lead to an increased or equal NPV

# Numerical example 1

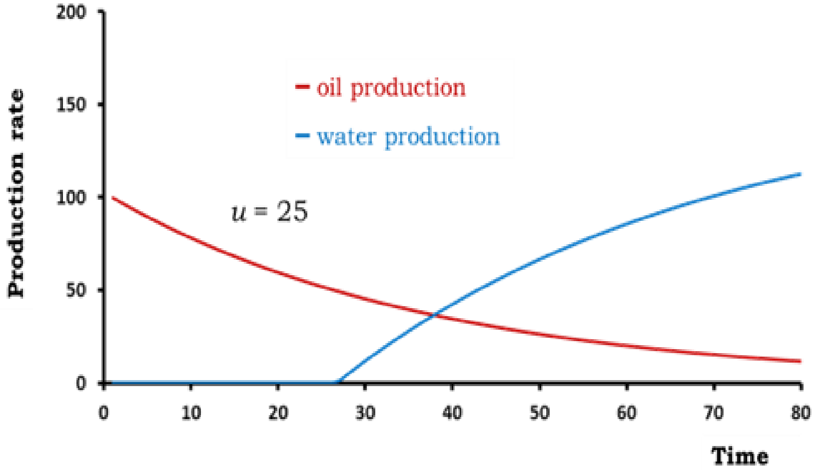
- Toy model based on 'decline curves'
- Scalar control variable: $u$

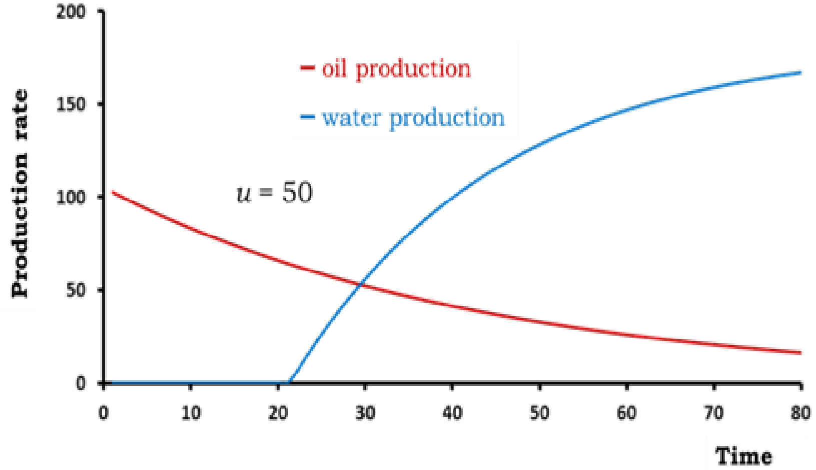
$$q_o(u,t) = \left(q_{o,ini} + c_1 u\right)\exp\left(-\frac{t}{a + \frac{1}{c_2}u}\right)$$

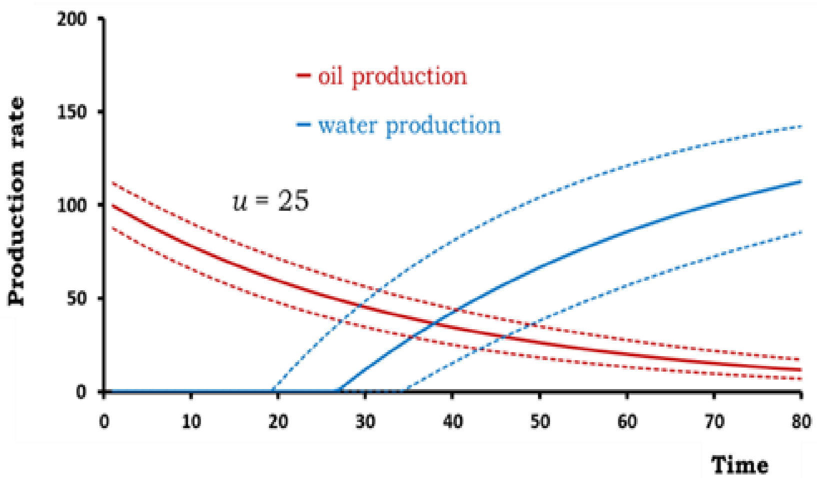
- Vector of uncertain parameters: $a, q_{o,ini}, q_{w,\infty}, t_{bt}$

$$q_w(u,t) = H\left[t_{bt}\left(1 - \frac{1}{c_3}u\right)\right]\left(q_{w,\infty} + u\right)\left[1 - \exp\left(-\frac{t - t_{bt}\left(1 - \frac{1}{c_3}u\right)}{c_4 a - \frac{1}{c_5}u}\right)\right]$$

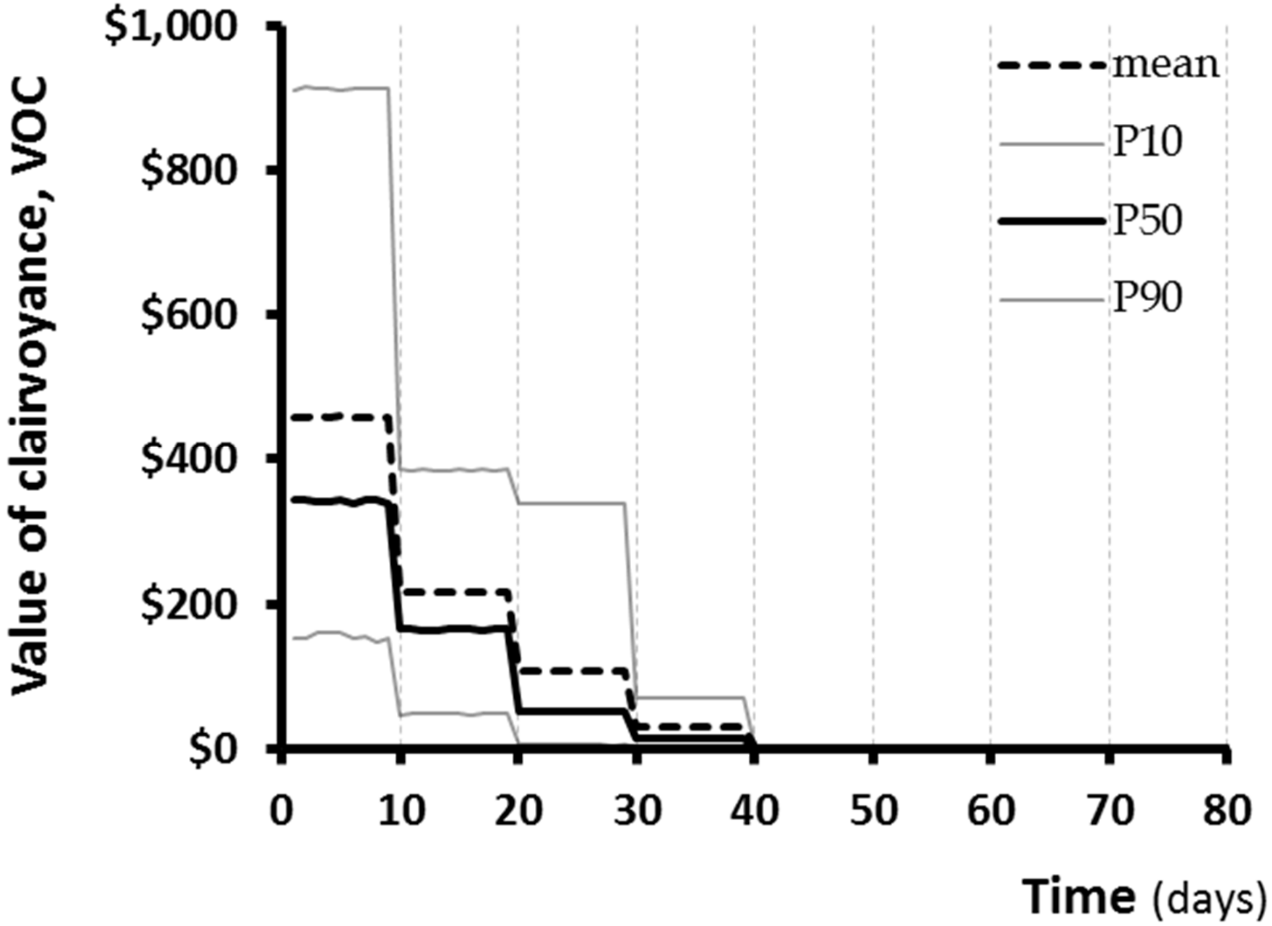
# Oil and water production example 1



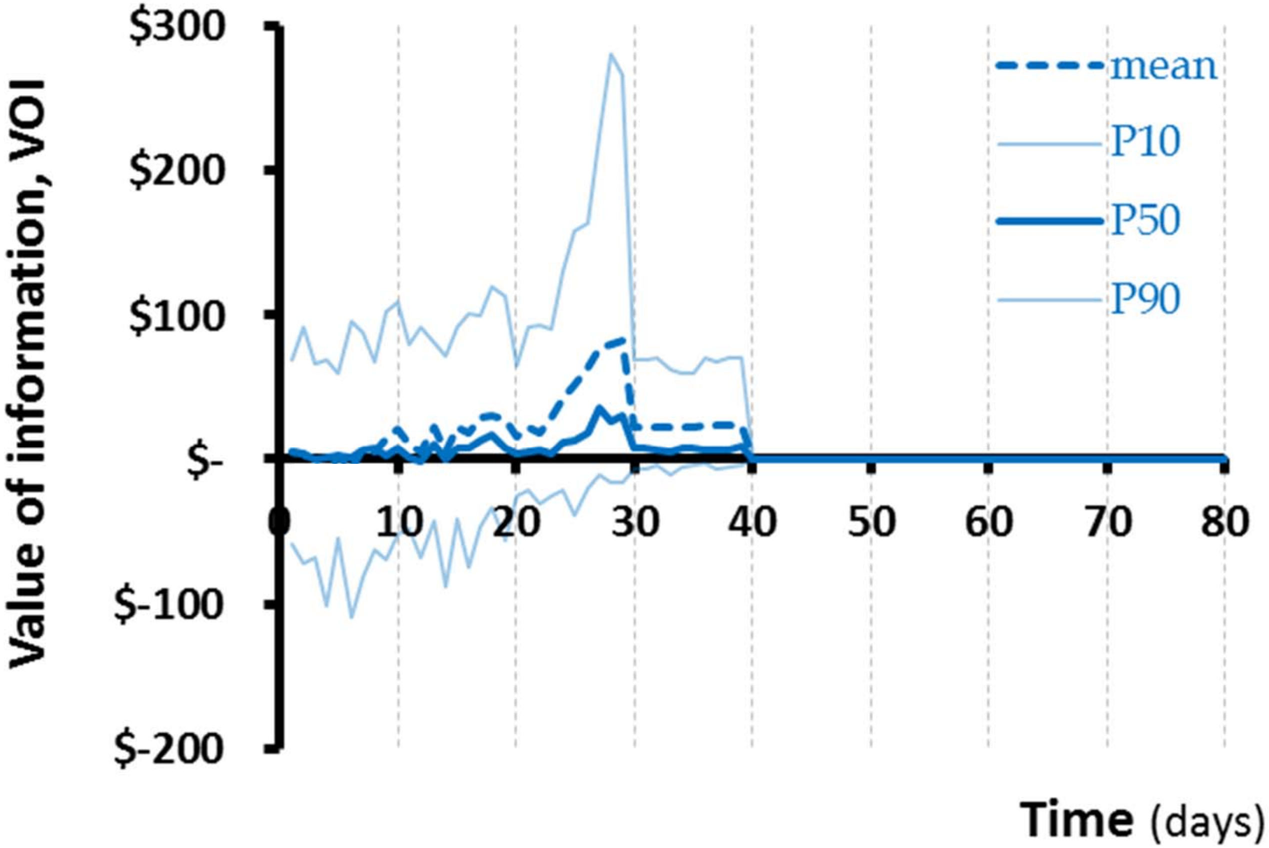



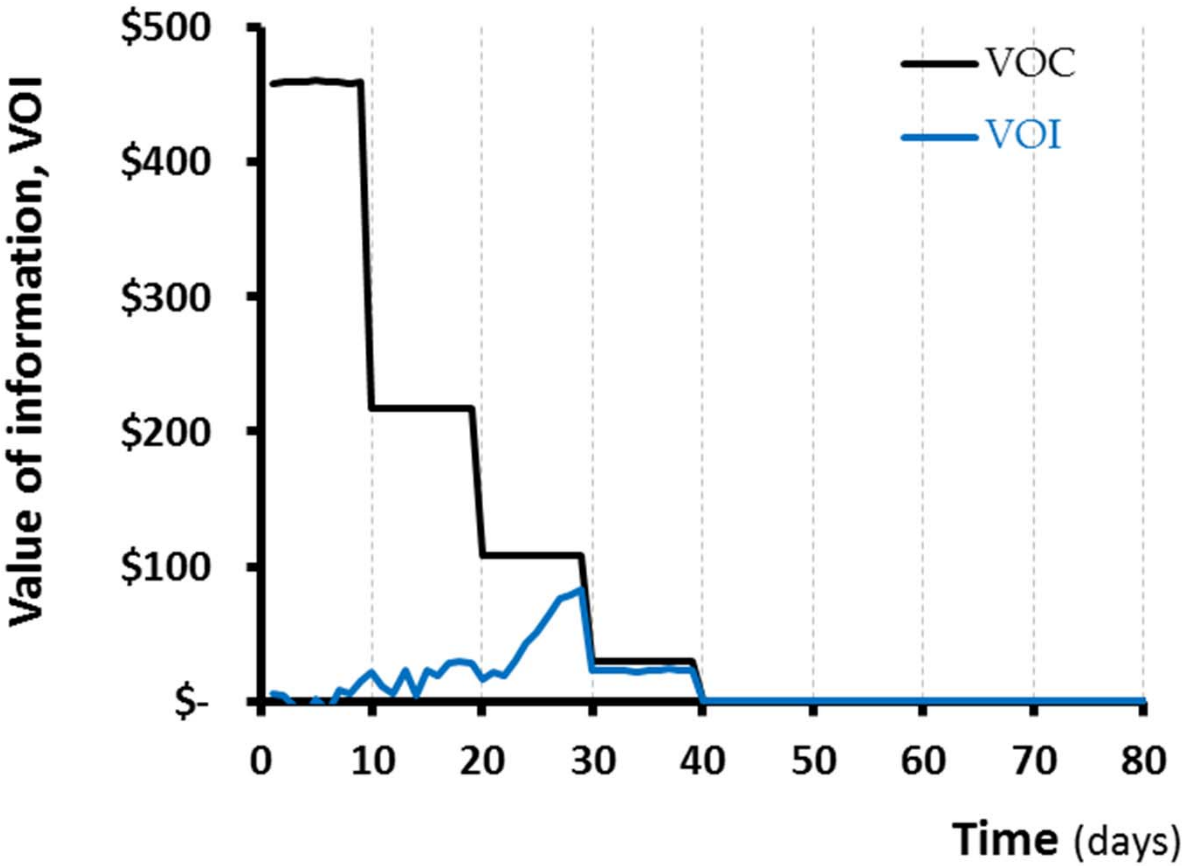
# Results example 1: VOC



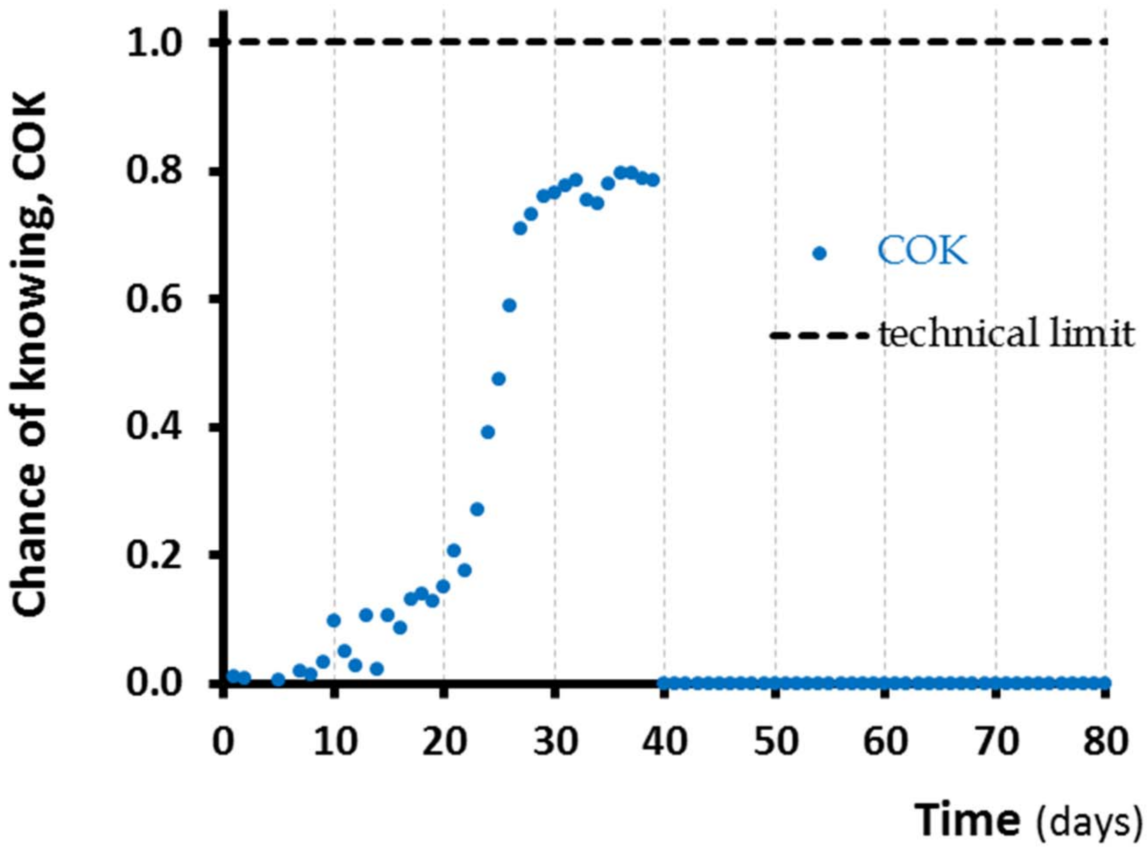
# Results example 1: VOI



# Results example 1:  VOC and VOI

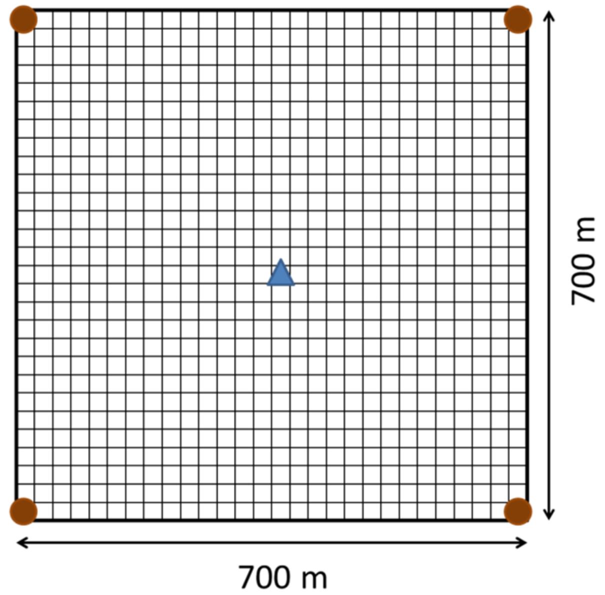


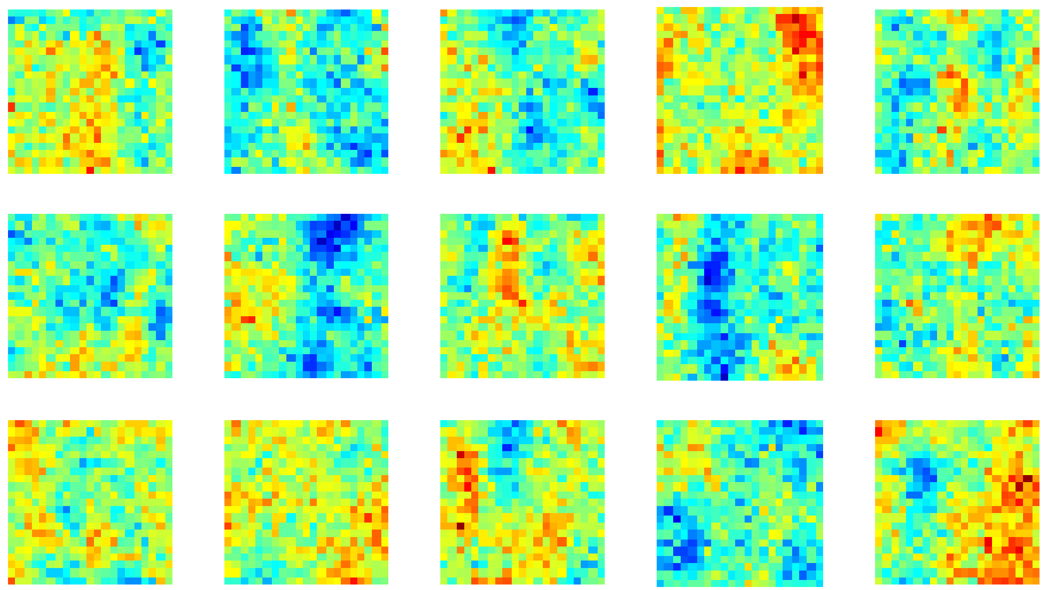
# Results example 1: COK = VOI / VOC



# Numerical example 2

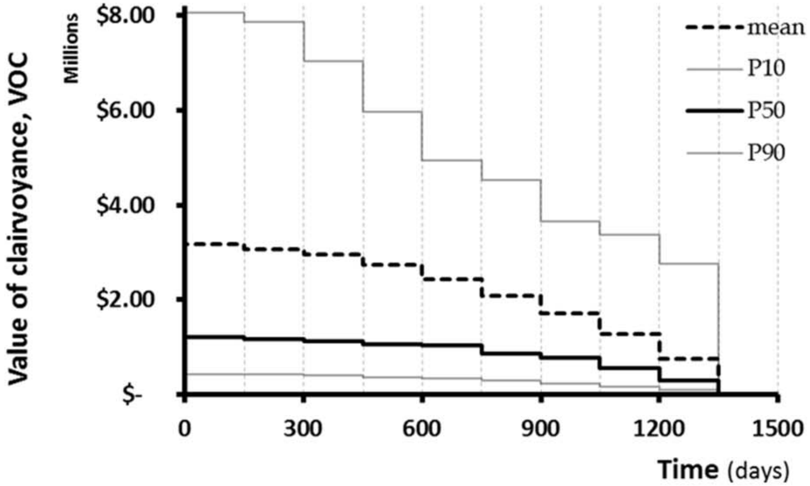
- ‘Inverted five spot’ configuration: one central water injector, four oil producers in the corners (top view)

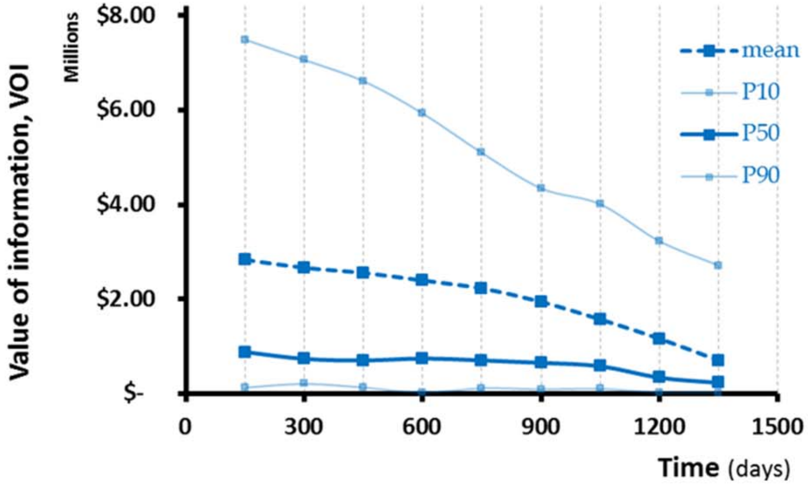


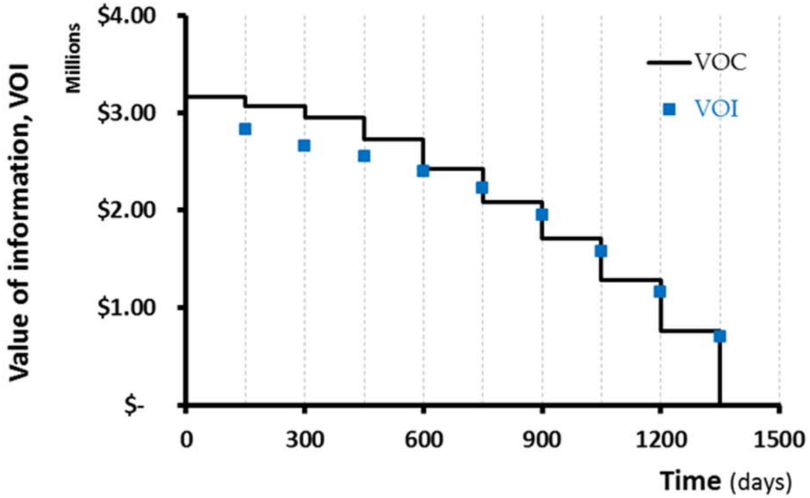



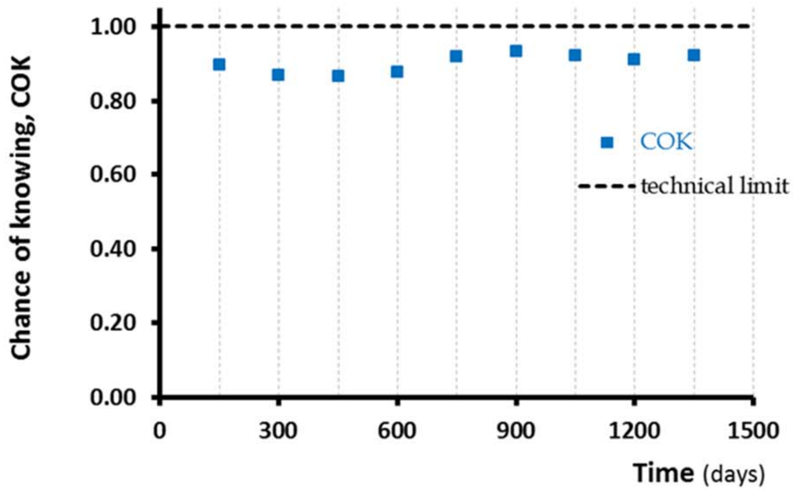
- 50 ensemble members (different porosity and perm. fields)

# Results example 2



# Conclusions

- VOI-for-CLRM method works well for examples so-far
- Computationally very intensive
- Next steps (ongoing):
  - Test method for assimilation at multiple times and for different data types
  - Test method on larger examples (using model-order reduction to achieve computational feasibility)
- Reference
  - Barros, E.G.D., Jansen, J.D. and Van den Hof, P.M.J., 2015: Value of information in closed-loop reservoir management. Accepted for publication in *Computational Geosciences*.

# Acknowledgments


- This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of TNO, Delft University of Technology, ENI, Statoil and Petrobras

- We used the Matlab Reservoir Simulation Toolbox (MRST), an open-source simulator developed by Sintef (Norway) which can be obtained from http://www.sintef.no/projectweb/mrst/

- The EnKF module for MRST was developed by Olwijn Leeuwenburgh (TNO) and can be obtained from http://www.isapp2.com/data-sharepoint/enkf-module-for-mrst.